Attorney Docket No. 81862.P186 *PATENT*

UNITED STATES PATENT APPLICATION

FOR

PROGRAMMABLE FSM ENGINE USING A NEW CAS APPLICATION LANGUAGE

First Named Inventor:
Kenneth Hyams

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1026
(408) 720-8300

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number EL 617 210 115 US

Date of Deposit December 22, 2000

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231

Tina Domingo
(Typed or printed name of person mailing paper or fee)

T Domingo 12-22-2000
(Signature of person mailing paper or fee) Date

PROGRAMMABLE FSM ENGINE USING A NEW CAS APPLICATION LANGUAGE

FIELD OF INVENTION

The present invention is related to the field of voice over internet protocol (VoIP) networks.

BACKGROUND OF THE INVENTION

Conventional networks route telephone signals over an internet protocol (IP) network by converting channel associated signals (CAS) into IP signals using a CAS device. The CAS device serves as a type of gateway between the public branch exchange (PBX) telephone system and the IP network. The CAS device is connected to one or more PBX systems through a line such as a T1 line, for example. The T1 line may connect the CAS device to multiple PBX telephone systems. Each PBX system may have its own specific CAS protocol. However, only one CAS protocol can be used with the line. In order to change the line's CAS protocol, the line must be taken out of service. Alternatively, the CAS module must be able to support a large number of CAS protocols.

It is extremely difficult to support many, if not all, CAS protocols using a single CAS module. Furthermore, if a given protocol is upgraded, the CAS module must be upgraded by taking the corresponding telephone lines out of service.

SUMMARY OF THE INVENTION

An apparatus including a private branch exchange (PBX) switch, a residential gateway to receive channel associated signals (CAS) from the PBX switch, and a programmable CAS module to convert the CAS signals into internet protocol (IP) signals is disclosed.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements, and in which:

Figure 1 shows an example of a network that includes a programmable CAS module.

Figure 2 shows an embodiment of residential gateway that includes programmable CAS module.

Figure 3 shows an embodiment of a programmable CAS module.

Figure 4 shows an example of an embodiment of a method to allow a user to program the CAS module.

DETAILED DESCRIPTION

A programmable finite state machine to process channel associated signals is disclosed. In one embodiment, the machine includes a private branch exchange (PBX) switch, a residential gateway to receive channel associated signals (CAS) from the PBX switch, and a programmable CAS module to convert the CAS signals into internet protocol (IP) signals.

Using the programmable CAS module, a new CAS protocol can be supported by the programmable module by writing an application program for the new CAS protocol and downloading the new protocol into the programmable CAS module. This eliminates the need for replacing the CAS module with a new module in order to support a different CAS protocol. Furthermore, given that there are innumerable PBX vendors, with different timing requirements for the CAS signals, changing the CAS application program to support a different vendor's timing signals can be done in the field, without having to go through a laborious release process.

Thus, a new CAS variant can be supported by downloading the new CAS variant protocol definition files into the programmable CAS module. The service provider can thus support new CAS variants without needing expensive hardware upgrades. Also, the release cycle is shorter and simpler in order to support a new CAS variant.

Figure 1 shows an example of a network that includes a programmable CAS module. As shown in **Figure 1,** a telephone call that originates at a given telephone 110 goes through a customer-premises telephony switch, such as a private branch exchange (PBX) 120, and may be routed through a residential gateway (RGW) 130, to enable voice transmission over an internet protocol (IP) network 100. The RGW 130 may engage the telephony 110 with the PBX equipment 120 through channelized T1 or E1 trunks 125. The RGW 130 processes channel associated signaling (CAS) information to identify fundamental circuit states of the telephony equipment to the IP network. For example, with a T1 trunk, the channel associated signaling used to establish and terminate calls, such as on-hook and off-hook signals, may be carried in the channel that will also carry the call data. For an E1 trunk, these CAS signals may be carried in a dedicated channel, while in-band signaling such as dual tone multi-frequency (DTMF) signals may be transmitted in the call channel. The RGW 130 receives these signals and converts them into IP signals, such as xGCP signals for example.

In one embodiment, the RGW converts the CAS signals into IP signals such as gateway control protocol (GCP) signals, for example, using programmable CAS module 135. The GCP signals may be media gateway control protocol (MGCP) signals or simple gateway control protocol (SGCP) signals, for example. The xGCP signals are sent to call agent 140 through IP

network 100. Call agent 140 identifies the appropriate telephone that should receive the telephone call that originated at telephone 110, and establishes a connection with the appropriate telephone 190 by sending an appropriate command to call agent 145 and residential gateway 150. Residential gateway 150 converts the xGCP signals back into CAS signals, and sends the CAS signals to the appropriate telephone 190 through PBX 170. The residential gateway 150 may be connected to PBX 170 through a T1 or E1 connection 155.

Figure 2 shows an embodiment of residential gateway 130 that includes programmable CAS module 135. The programmable CAS module 135 sends and receives the CAS signals to other components of the RGW. The call control device 220 sends and receives the xGCP messages to and from call agent 140. The call control device decodes and demultiplexes xGCP messages that it receives from the call agent over the IP network and outputs corresponding CAS signals to CAS module 135. The call control device 220 also receives CAS signals, converts them into xGCP, and transmits them to the call agent. This allows for delivery of message components to their appropriate recipient.

The call control device 220 informs the CAS module 135 of simple gateway control protocol (SGCP) notification requests received from the call agent. The notification request may be a standalone message, or may be part of connection oriented messages for the call control device. The CAS module 135 receives indications of signals received from the PBX as well as signals to the

PBX through signaling driver subsystem 230. Indications of signals include changes in signaling (A, B, C, D) bits for events such as off-hook, receipt of dial digits, DTMF, MF, and pulse.

The management module 260 may be used to program or define parameters of the CAS module 135. Thus, the CAS module 135 may be a finite state machine (FSM) that is a programmable CAS state machine (PCSM). For example, an octet string upon which a structure is placed to comprise various tables may be loaded into the CAS module by the management subsystem. However, other methods of downloading the application to the CAS engine may be used. This may be obtained by command line interpretation, SNMP or other protocols from remote sources such as the call agent. A table, or data structure, contained in the octet string may define states, events, timing durations, and actions for a PCSM. Other tables may define line signal bits and durations, tone frequencies, cadences and durations, and general parameters related to the connected PBX and specific CAS protocol that influences the behavior of the PCSMs.

The CAS module may be generic to allow support for different CAS protocols. The generic CAS module can handle a specific CAS protocol using programmable state machine files that may be downloaded into the RGW. The downloaded files may contain both the signal definitions as well as the state/event/action data for a given CAS protocol.

Figure 3 shows an embodiment of a programmable CAS module 135. There may be three finite state machines on the RGW host processor that process the CAS information. The local call agent (LCA) FSM 320 interfaces with the call control device to handle CAS event requests from the call agent. The LCA FSM 320 also interfaces with the receive line signal definition (Rx LSD) FSM 310 and the transmit (Tx) LSD FSM 330 to communicate with the PBX. The LCA FSM 320 interprets user defined state/event/action and gives appropriate responses based on the received event for the current state. The LCA FSM may be implemented as a processor which decodes the opcode user-defined actions specified in the CAS file and executes them.

Any CAS events observed on an endpoint are received by the Rx Line Signal Definition (LSD) FSM 310. The Rx LSD FSM 310 informs the LCA 320 about the interpreted event based on the downloaded incoming signal definition. The call agent is then notified of the event through the call control device.

The Rx LSD FSM 310 recognizes changes in the state of line signaling (A, B, C, D bits), as well as inband tones. When a change in signalling bits is recognized, the Rx LSD FSM 310 searches a table for a match to the former state and new state. That table entry then indicates the longest minimum duration for which that pattern constitutes a particular signal.

The Rx LSD FSM 310 sets a timer for that period and waits. If the timer expires, it has seen the signal so it sends an indication to the Local Call Agent FSM 320; it then awaits notification of another signal level change. Should the timer not expire before another signal level change occurs, the Rx LSD FSM 310 checks whether the signal has returned to its original state. If so, this may be a pulse of some known type. Table entries adjacent to the one it had been using may contain the same pattern as seen but with minimum and maximum durations. If one of these matches, it sends an indication to the LCA FSM 320. Finally, if the signal state does not match the original signal state, there was not a pulse.

In its idle state, the Tx LSD FSM 330 waits for a request from the LCA 320 to send a signal out to the PBX. Signals are applied by making a request to the Signaling Driver. For tones or dial pulses there may be parameters found in a table to include in the request. For an A-B bit signals there may be a duration associated which will also be found in a table. The Tx LSD FSM 330 may time the signal, returning the levels to their former state when the timer expires. The Tx LSD FSM 330 returns to the LCA 320 a confirmation that the signal request was completed, or not. If not, it makes a callback later to the LCA indicating completion of the signal request. For A-B bits this may occur when the timer expires for a pulsed signal. For timed tones (such as dial digits, whether tones or pulses) this may occur when a message arrives from the Signaling Driver

indicating completion. Thus, the Tx LSD FSM is responsible for transmitting CAS signals to the PBX.

A user may program the portions of the LCA 320 which respond to signals from the Rx LSD FSM 310, timer expirations, and may program messages from the remote call agent and signaling requested therein. Programmable cells for incoming signaling events may have an associated state. There may be many cells for the same state, responding to different events.

Thus, the CAS module can be used with different CAS protocols. The user may provide definitions for incoming and outgoing signals along with the state/event/action data for a given CAS protocol using a user interface tool. The user interface tool may run on a workstation and may create a CAS program file which includes a system section, an incoming signals section, an outgoing signals section, and a state events action section. This CAS program file may be compiled by a CAS compiler to generate a binary CAS file. A given variety of CAS protocol may have a binary CAS file that contains the given protocol specific signal definitions and actions pertaining to the given CAS protocol.

The associated primitives may be provided to the user in order to generate this file. The user interface collects the user data and generates the CAS file. The generated file may be downloaded to the programmable CAS module. Subsequently, a CAS parser program then parses the CAS file and

creates data structures corresponding to the signal definitions and state/event/action data. Using the user interface and the primitive functions, the user is thus able to define the CAS information needed by the CAS module for the given CAS protocol.

Figure 4 shows an example of an embodiment of a method to allow a user to program the CAS module. The method includes allowing a user to define a state, an event, or an action of a telephony protocol, 410, downloading the user defined state, event, or action to a channel associated signal (CAS) engine, 420, and changing a telephony protocol of the CAS engine corresponding to idle telephone lines associated with the CAS engine based on the downloaded user defined state, event, or action, 430. The the telephony protocol may be a CAS protocol. The state is a transient condition of the CAS engine, such as "idle," or "originate an outgoing call," for example. The event is an external trigger received by the CAS engine, such as "receive off-hook," or "receive digits," for example. The action is a response by the CAS engine to a state-event condition, such as "given a certain state and a certain event, perform this action." The action may be "initiate transmitter," or "notify call agent," for example. The method of **Figure 4** may be performed by a processing system that reads computer instructions from memory. The computer instructions, when executed by the system, may cause the system to perform the method of **Figure 4.** The method of **Figure 4** may also be performed by logic devices, such

as programmable logic devices, configured to implement the method to allow the user to program the CAS module.

These and other embodiments of the present invention may be realized in accordance with the following teachings and it should be evident that various modifications and changes may be made in the following teachings without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than restrictive sense and the invention measured only in terms of the claims.